7

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/839,045	04/20/2001	Dietrich Charisius	30013630-0008	3644

23485 7590 05/12/2004

JINAN GLASGOW
P O BOX 28539
RALEIGH, NC 276118539

EXAMINER	
INGBERG, TODD D	
ART UNIT	PAPER NUMBER
2124	10

DATE MAILED: 05/12/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

4

Office Action Summary	**Application No.** 09/839,045	**Applicant(s)** CHARISIUS ET AL.	
	Examiner Todd Ingberg	**Art Unit** 2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *22 March 2004*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-183* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-183* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date 3,5.

4) □ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

Claims 1 – 183 have been examined.

Specification

1. Pages 1 and 2 of the Specification have a Cross –Reference to Related Application section. Examiner is requesting the Applicant complete the missing **case numbers** and submit this revision with the response to this office action.

Priority

2. Claim to Priority is still being investigated.

Information Disclosure Statement

3. The information disclosure statement filed July 16, 2001 and September 10, 2001 have been considered.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 1- 183 are rejected under 35 U.S.C. 102(b) as being anticipated by Template Software's commercial product line.

The **Template** product line contains:

The SNAP programming language (Not used in this Office Action)

The Workflow Template (Two manuals used)

The Web Component (Not used in this Office Action)

These three layered products work together.

The documentation sets for the products contains the following manuals.

SNAP released June 1997

SNAP Language Reference (Not used in this Office Action)

Using the SNAP Language (Not used in this Office Action)

Using the SNAP Communication Component (Not used in this Office Action)

Using the SNAP Graphic User Interface Component (Not used in this Office Action)

Getting Started with SNAP (Not used in this Office Action)

Using the SNAP Display Editors (Not used in this Office Action)

SNAP Class Library Reference (Not used in this Office Action)

Using the SNAP External Application Software Component (Not used in this Office Action)

Using the SNAP Development Environment (Referred to as **SNAP**)

SNAP Module Library Reference (Not used in this Office Action)

Using the SNAP Permanent Storage Component (Not used in this Office Action)

Workflow released September 1997

Developing a WFT Workflow System (Not used in this Office Action)

Using the WFT Development Environment (Not used in this Office Action)

WFT Library Reference (Not used in this Office Action)

Web Component

Using the Web Component (Not used in this Office Action)

Since, these products work together they constitute a single reference and can be used as the basis for a rejection based on anticipated by a product offering. Furthermore, with the 1997 press release announcing version 8.0 these considered prior art under ***In re Epstein*** 31 USPQ2d 1817 (decided August 17, 1994) with a 1997 release date despite the 1998 copyright date.

SNAP anticipates an object oriented programming environment (OO-CASE tool) with an object model editor (**SNAP**, page 3-6). With in the Object model editor the user is able to link classes. The links include inheritance and relation lines (**SNAP**, page 3-9). SNAP support a plurality of dependencies (**SNAP**, page 3-14). In addition to the lines representing a form of interface the functions also interface (page 3-44 parameters). One of ordinary skill in the art should know messaging is inherent in object technology is also a form of interface (SNAP, page 3-44 functions are methods also can be called packages which support method calls). Furthermore, the SNAP environment provides for error messages when rules in the object model editor have been violated (SNAP, pages 2-15 and 2-17). SNAP also supports a class (node) representing multiple links (SNAP, page 3-15 down arrow indicating child classes are present but not shown). The ability to declare variables (attributes) is preformed by the Attribute tab (SNAP, page 3-40). The code between two classes linked through inheritance is performed dynamically and is visible the class editor (SNAP, page 3-31 – Inherits From). It should be noted that the same interpreted environment generates the error messages above. Also the inherited attributes and class attributes are visible on SNAP page 3-40. Local attributes to functions/packages/methods are shown on SNAP page 3-44

Conclusion

6. ***Undue multiplicity*** is an unpopular rejection within the Patent Office at this time. However, the Examiner did not see the value in 183 claims. The claims are repeated below.

Claim 1
A method in a data processing system having a plurality of elements, each element having corresponding code, the method comprising the steps of: receiving a request to form a link; receiving an indication of a first of the plurality of elements; receiving an indication of a second of the plurality of elements; and in response to receiving the request, the indication of the first element, and the indication of the second element, adding new code to the first element to reflect the link to the second element.

Claim 2
The method of claim 1, further comprising the step of displaying a graphical representation of the code associated with the first element.

Claim 3
The method of claim 2, further comprising the step of modifying the graphical representation of the code associated with the first element to reflect the link to the second element.

Claim 4
The method of claim 1, wherein the step of adding new code to the first element comprises the steps of: determining whether linking the first element to the second element would violate a predefined rule; and when it is determined that linking the first element to the second element would not violate a predefined rule, adding the new code to the first element to form the link to the second element.

Claim 5
The method of claim 4, wherein the step of determining whether linking the first element to the second element would violate a predefined rule comprises the steps of determining whether the first element is a class and whether the second element is another class; and when it is determined that the first element is the class and that the second element is the other class, identifying the link from the first element to the second element as an inheritance link.

Claim 6
The method of claim 5, further comprising the step of identifying a link error when it is determined that the first element is the class and that the second element is not the other class.

Claim 7
The method of claim 4, wherein the step of determining whether linking the first element to the second element would violate a predefined rule comprises the steps of determining whether the first element is a class and whether the second element is an interface; and when it is determined that the first element is the class and that the second element is the interface, identifying the link from the first element to the second element as an implementation link.

Claim 8
The method of claim 7, further comprising the step of identifying a link error when it is determined that the first element is the class and that the second element is not the interface.

Claim 9
The method of claim 4, wherein the step of determining whether linking the first element to the second element would violate a predefined rule comprises the steps of: determining whether the first element is an interface and the second element is another interface; and when it is determined that the first element is the interface and the second element is the other interface, identifying the link from the first element to the second element as an inheritance link.

Claim 10
The method of claim 9, further comprising the step of identifying a link error when it is determined that the first element is the interface and the second element is not the other interface.

Claim 11
A method in a data processing system having a plurality of elements, each element having corresponding code, the method comprising the steps of: receiving a request to form a link; receiving an indication of a first of the plurality of elements; receiving an indication of a second of the plurality of elements; determining whether linking the first element to the second element would violate a predefined rule; and when it is determined that linking the first element to the second element would not violate a predefined rule, adding new code to the first element to reflect the link to the second element.

Claim 12
The method of claim 11, further comprising the step of displaying a graphical representation of the code associated with the first element.

Claim 13
The method of claim 12, further comprising the step of modifying the graphical representation of the code associated with the first element to reflect the link to the second element.

Claim 14
The method of claim 11, wherein the step of determining whether linking the first element to the second element would violate a predefined rule comprises the steps of determining whether the first element is a class and whether the second element is another class; and when it is determined that the first element is the class and that the second element is the other class, identifying the link from the first element to the second element as an inheritance link.

Claim 15
The method of claim 11, wherein the step of determining whether linking the first element to the second element would violate a predefined rule comprises the steps of: determining whether the first element is a class and whether the second element is an interface; and when it is determined that the first element is the class and that the second element is the interface, identifying the link from the first element to the second element as an implementation link.

Claim 16

The method of claim 11, wherein the step of determining whether linking the first element to second element would violate a predefined rule comprises the steps of: determining whether the first element is an interface and the second element is another interface; and when it is determined that the first element is the interface and the second element is the other interface, identifying the link from the first element to the second element as an inheritance link.

Claim 17
A method in a data processing system having a plurality of elements, each element having corresponding code, the method comprising the steps of: displaying a graphical representation of the code associated with a first of the plurality of elements and a graphical representation of the code associated with a second of the plurality of elements; receiving a request to form a link from the first element to the second element; determining whether the first element is a class and whether the second element is another class; and when it is determined that the first element is the class and that the second element is the other class, identifying the link from the first element to the second element as an inheritance link; adding new code to the first element to reflect the link to the second element; and modifying the graphical representation of the code associated with the first element to reflect the link to the second element.

Claim 18
The method of claim 17, further comprising the steps of: when it is determined that the first element is the class and that the second element is not the other class, determining whether the second element is an interface; and when it is determined that the second element is the interface, identifying the link from the first element to the second element as an implementation link.

Claim 19
The method of claim 17, further comprising the steps of: when it is determined that the first element is not the class and that the second element is not the other class, determining whether the first and the second elements are interfaces; and when it is determined that the first and second elements are interfaces, identifying the link from the first element to the second element as an inheritance link.

Claim 20
A method in a data processing system having a plurality of elements and having a link between two of the plurality of elements, wherein each element has corresponding code and the linked elements include a source and a destination, the method comprising the steps of: receiving a selection of one of the linked elements; receiving an identification of another of the plurality of elements that is different than the linked elements; determining whether the selected element is the destination; and when it is determined that the selected element is the destination, modifying the corresponding code of the other element to reflect a new link between the other element and the destination element.

Claim 21

The method of claim 20, wherein the modifying step further includes the step of modifying the corresponding code of the source to reflect the removal of the link between the source and the destination.

Claim 22
The method of claim 21, further comprising the step of displaying a graphical representation of the corresponding code of the source.

Claim 23
The method of claim 22, further comprising the step of modifying the graphical representation of the corresponding code of the source to reflect the removal of the link between the source and the destination.

Claim 24
The method of claim 20, further comprising the step of displaying a graphical representation of the corresponding code of the other element.

Claim 25
The method of claim 24, further comprising the step of modifying the graphical representation of the corresponding code of the other element to reflect the new link between the other element and the destination element.

Claim 26
The method of claim 20, wherein the modifying step includes the steps of: determining whether linking the other element to the destination would violate a predefined rule; and when it is determined that linking the other element to the destination would not violate a predefined rule, modifying the corresponding code of the source to reflect the removal of the link between the source and the destination; and adding new code to the corresponding code of the other element to reflect the new link between the other element and the destination element.

Claim 27
The method of claim 26, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of: determining whether the other element is a class and whether the destination is another class; and when it is determined that the other element is the class and that the destination is the other class, identifying the new link between the other element and the destination as an inheritance link.

Claim 28
The method of claim 26, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of determining whether the other element is a class and whether the destination is an interface; and when it is determined that the other element is the class and that the destination is the interface, identifying the new link between the other element and the destination as an implementation link.

Claim 29
The method of claim 28, further comprising the step of identifying a link error when it is determined that the other element is the class and that the destination is not the interface.

Claim 30
The method of claim 26, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of: determining whether the other element is an interface and the destination is another interface; and when it is determined that the other element is the interface and the destination is the other interface, identifying the new link between the other element and the destination as an inheritance link.

Claim 31
The method of claim 30, :further comprising the step of identifying a link error when it is determined that the other element is not the interface.

Claim 32
The method of claim 30, further comprising the step of identifying a link error when it is determined that the destination is not the other interface.

Claim 33
The method of claim 20 further comprising the step of when it is determined that the selected element is the source, modifying the corresponding code of the source to reflect a new link between the source and the other element.

Claim 34
The method of claim 33, further comprising the steps of: when it is determined that the selected element is the source, determining whether linking the source to the other element would violate a predefined rule; and when it is determined that linking the source to the other element would not violate a predefined rule, modifying the corresponding code of the source to reflect the removal of the link between the source and the destination; and adding new code to the corresponding code of the source to reflect the new link to the other element.

Claim 35
The method of claim 34, further comprising the step of displaying a graphical representation of the code of the source to reflect the removal of the link and to reflect the new link to the other element when it is determined that the selected element is the source.

Claim 36
The method of claim 34, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of: determining whether the source is a class and whether the other element is another class; and when it is determined that the source is the class and that the other element is the other class, identifying the new link between the source and the other element as an inheritance link.

Claim 37
The method of claim 34, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of: determining whether the source is a class and whether the other element is an interface; and when it is determined that the source is the class and that the other element is the interface, identifying the new link from the source to the other element as an implementation link.

Claim 38
The method of claim 37, further comprising the step of identifying a link error when it is determined that the other element is not the interface.

Claim 39
The method of claim 34, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of: determining whether the source is an interface and the other element is another interface; and when it is determined that the source is the interface and the other element is the other interface, identifying the new link between the source and the other element as an inheritance link.

Claim 40
The method of claim 39, further comprising the step of identifying a link error when it is determined that the source is not the interface.

Claim 41
The method of claim 39, further comprising the step of identifying a link error when it is determined that the other element is not the other interface.

Claim 42
A method in a data processing system having a plurality of elements and having a link between two of the plurality of elements, wherein each element has corresponding code and the linked elements include a source and a destination, the method comprising the steps of: receiving an identification of the link; receiving a selection of one of the linked elements; receiving an identification of another of the plurality of elements that is different than the linked elements; determining whether the selected element is the source; and when it is determined that the selected element is the source, modifying the corresponding code of the source to reflect a new link between the source and the other element.

Claim 43
The method of claim 42, further comprising the step of displaying a graphical representation of the corresponding code of the source and a graphical representation of the corresponding code of the other element.

Claim 44

The method of claim 43, further comprising the step of modifying the graphical representation of the corresponding code of the source to reflect the removal of the link when it is determined that the selected element is the source.

Claim 45
The method of claim 42, further comprising the steps of: when it is determined that the selected element is the source, determining whether linking the source to the other element would violate a predefined rule; and when it is determined that linking the source to the other element would not violate a predefined rule, modifying the corresponding code of the source to reflect the removal of the link between the source and the destination; and adding new code to the corresponding code of the source to reflect the new link to the other element.

Claim 46
The method of claim 45, further comprising the step of displaying a graphical representation of the code of the source to reflect the removal of the link and to reflect the new link to the other element when it is determined that the selected element is the source.

Claim 47
The method of claim 45, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of determining whether the source is a class and whether the other element is another class; and when it is determined that the source is the class and that the other element is the other class, identifying the new link between the source and the other element as an inheritance link.

Claim 48
The method of claim 45, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of: determining whether the source is a class and whether the other element is an interface; and when it is determined that the source is the class and that the other element is the interface, identifying the new link atom the source to the other element as an implementation link.

Claim 49
The method of claim 48, further comprising the step of identifying a link error when it is determined that the other element is not the interface.

Claim 50
The method of claim 45, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of determining whether the source is an interface and the other element is another interface; and when it is determined that the source is the interface and the other element is the other interface, identifying the new link between the source and the other element as an inheritance link.

Claim 51

The method of claim 50, further comprising the step of identifying a link error when it is determined that the source is not the interface.

Claim 52
The method of claim 50, further comprising the step of identifying a link error when it is determined that the other element is not the other interface.

Claim 53
The method of claim 42 further comprising the step of: when it is determined that the selected element is the destination, modifying the corresponding; code of the other element to reflect a new link between the other element and the destination element.

Claim 54
The method of claim 53, wherein the modifying step further includes the steps of: determining whether linking the other element to the destination would violate a predefined rule; and when it is determined that linking the other element to the destination would not violate a predefined rule, modifying the corresponding; code of the source to reflect the removal of the link between the source and the destination; and adding new code to the corresponding code of the other element to reflect a new link between the other element and the destination element.

Claim 55
The method of claim 54, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of determining whether the other element is a class and whether the destination is another class; and when it is determined that the other element is the class and that the destination is the other class, identifying the new link between the other element and the destination as an inheritance link.

Claim 56
The method of claim 54, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of determining whether the other element is a class and whether the destination is an interface; and when it is determined that the other clement is the class and that the destination is the interface, identifying the new link between the other element and the destination as an implementation link.

Claim 57
The method of claim 56, further comprising the step of identifying a link error when it is determined that the destination is not the interface.

Claim 58
The method of claim 54, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of: determining whether the other element is an interface and the destination is another interface; and when it is determined that the other element is the interface and the destination is the other interface, identifying the new link between the other element and the destination as an inheritance link.

Claim 59
The method of claim 58, further comprising the step of identifying a link error when it is determined that the other element is not the interface.

Claim 60
The method of claim 58, further comprising the step of identifying a link error when it is determined that the destination is not the other interface.

Claim 61
A method in a data processing system having a plurality of elements and having a link between two of the plurality of elements, wherein each element has corresponding code and the linked elements include a source and a destination, the method comprising the steps of: displaying a graphical representation of the corresponding code of each of the plurality of elements; receiving a selection of one of the linked elements; receiving an identification of another of the plurality of elements that is different than the linked element; determining whether the selected element is the destination; and when it is determined that the selected element is the destination, determining whether the other element is a class and whether the destination is another class; and when it is determined that the other element is the class and that the destination is the other class, identifying a new link from the other element to the destination as an inheritance link; removing a portion of the corresponding code of the source that reflects the link between the source and the destination; adding new code to corresponding code of the other element to reflect the new link between the other element and the destination; modifying the graphical representation of the corresponding code of the source to reflect the removal of the link; and modifying the graphical representation of the corresponding code of the other element to reflect the new link.

Claim 62
The method of claim 61, further comprising the steps of: when it is determined that the other element is the class and that the destination is not the other class, determining whether the destination is an interface; and when it is determined that the destination is the interface, identifying the new link between the other element and the destination as an implementation link.

Claim 63
The method of claim 61, further comprising the steps of: when it is determined that the other element is not the class and that the destination is not the other class, determining whether the other element is an interface and whether the destination is another interface; and when it is determined that the other element is the interface and that the destination is the other interface, identifying the new link between the other element and the destination as an inheritance link.

Claim 64
The method of claim 61, further comprising the steps of: when it is determined that the selected element is the source, determining whether the source is a class and whether the other element is another class; and when it is determined that the source is the class and that the other element is

the other class, identifying the new link between the source and the other element as an inheritance link; removing a portion of the corresponding code of the source that reflects the link between the source and the destination; and adding new code to corresponding code of the source to reflect the new link between the source and the other element.

Claim 65
The method of claim 64, further comprising the step of modifying a graphical representation of the corresponding code of the source to reflect the removal of the link and to reflect the new link when it is determined that linking the source to the other element would not violate a predefined rule.

Claim 66
The method of claim 64, further comprising the steps of: when it is determined that the source is the class and that the other element is not the other class, determining whether the other element is in an interface; and when it is determined that the other element is the interface, identifying the new link between the source and the other element as an implementation link.

Claim 67
The method of claim 64, further comprising the steps of: when it is determined that the source is not the class and that the other element is not the other class, determining whether the source is an interface and the other element is another interface; and when it is determined that the source is the interface and the other element is the other interface, identifying the new link between the source and the other element as an inheritance link.

Claim 68
A method in a data processing system having a plurality of elements, the method comprising the steps of: receiving an identification of a first of the plurality of elements; receiving an identification of a second of the plurality of elements; receiving an indication that the first element is to be included in the second element; determining whether the first element is a class and whether the second element is another class; and when it is determined that the first element is the class and that the second element is the other class, transferring code corresponding to the first element into the second element.

Claim 69
The method of claim 68, wherein the method further comprises the step of displaying a graphical representation of the code of the first element and a graphical representation of the code of the second element.

Claim 70
The method of claim 69, further comprising the step of modifying the graphical representation of the code of the second element to reflect the transfer of the code corresponding to the first element into the second element.

Claim 71

The method of claim 68, wherein the step of transferring code comprises the steps of: removing the code corresponding to the first element from a file; placing the code corresponding to the first element within the code corresponding to the second element; and deleting the file.

Claim 72
The method of claim 68, wherein the method further comprises the steps of: when it is determined that the first element is the class and that the second element is not the other class, determining whether the second element is a package; and when it is determined that the second element is a package, moving a file that includes code corresponding to the first element to a directory associated with the second element.

Claim 73
A method in a data processing system having a plurality of elements, each element having corresponding code, the method comprising the steps of displaying a graphical representation of the code of a first of the plurality of elements and a graphical representation of the code of a second of the plurality of elements; receiving an indication that the first element is to be included in the second element; determining whether the first element is a class and whether the second element is another class; and when it is determined that the first element is the class and that the second element is the other class, transferring code corresponding to the first element into the second element; and modifying a graphical representation of the code of the second element to reflect the transfer of the first element into the second element.

Claim 74
The method of claim 73, wherein the step of transferring code comprises the steps of: removing the code corresponding to the first element from a file; placing the code corresponding to the first element within code corresponding to the second element; and deleting the file.

Claim 75
A method in a data processing system having a plurality of elements, each element having corresponding code, wherein code corresponding to a first of the plurality of elements is nested in the code corresponding to a second of the plurality of elements, the method comprising the steps of receiving an indication that the first element is to be removed from the second element; determining whether the first element is a class and whether the second element is another class; and when it is determined that the first element is the class and that the second element is the other class, removing code corresponding to the first element from the second element.

Claim 76
The method of claim 75, further comprising the step of placing the code corresponding to the first element into a file.

Claim 77
The method of claim 76, further comprising the step of displaying a graphical representation of the code corresponding to the first element.

Claim 78
The method of claim 75, further comprising the step of modifying a graphical representation of the code corresponding to the second element to reflect the removal of the first element from the second element.

Claim 79
The method of claim 75, further comprising the steps of when it is determined that the first element is the class and that the second element is not the other class, determining whether the second element is a package; and when it is determined that the second element is the package, removing a first file that includes code corresponding to the first element from a directory associated with the second element to another directory.

Claim 80
A method in a data processing system having a plurality of elements, wherein a first of the plurality of elements is nested within a second of the plurality of elements, the method comprising the steps of: receiving an indication that the first element is to be removed from the second element; determining whether the first element is a class and whether the second element is another class; and when it is determined that the first element is the class and that the second element is the other class, removing code corresponding to the first element from the second element; placing the code corresponding to the first element into a file; and displaying a graphical representation of the code corresponding to the first element.

Claim 81
The method of claim 80, further comprising the step of modifying a graphical representation of the code corresponding to the second element to reflect the removal of the first element from the second element.

Claim 82
The method of claim 80, further comprising the steps of: when it is determined that the first element is the class and the second element is not the other class, determining whether the second element is a package; and when it is determined that the second element is the package, removing a first file, that includes code corresponding to the first element from a directory associated with the second element.

Claim 83
A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of elements, each element having corresponding code, the method comprising the steps of: receiving a request to form a link; receiving an indication of a first of the plurality of elements; receiving an indication of a second of the plurality of elements; and in response to receiving the request, the indication of the first element, and the indication of the second element, adding new code to the first element to reflect the link to the second element.

Claim 84

The computer-readable medium of claim 83, wherein the method further comprises the step of displaying a graphical representation of the code associated with the first element.

Claim 85
The computer-readable medium of claim 84, wherein the method further comprises the step of modifying the graphical representation of the code associated with the first element to reflect the link to the second element.

Claim 86
The computer-readable medium of claim 83, wherein the step of adding new code to the first element comprises the steps of: determining whether linking the first element to the second element would violate a predefined rule; and when it is determined that linking the first element to the second element would not violate a predefined rule, adding the new code to the first element to form the link to the second element.
Claim 87
The computer-readable medium of claim 86, wherein the step of determining whether linking the first element to the second element would violate a predefined rule comprises the steps of: determining whether the first element is a class and whether the second element is another class; and when it is determined that the first element is the class and that the second element is the other class, identifying the link from the first element to the second element as an inheritance link.

Claim 88
The computer-readable medium of claim 87, wherein the method further comprises the step of identifying a link error when it is determined that the first element is the class and that the second element is not the other class.

Claim 89
The computer-readable medium of claim 86, wherein the step of determining whether linking the first element to the second element would violate a predefined rule comprises the steps of: determining whether the first element is a class and whether the second element is an interface; and when it is determined that the first element is the class and that the second element is the interface, identifying the link from the first element to the second element as an implementation link.

Claim 90
The computer-readable medium of claim 89, wherein the method further comprises the step of identifying a link error when it is determined that the first element is the class and that the second element is not the interface.

Claim 91
The computer-readable medium of claim 86, wherein the step of determining whether linking the first element to the second element would violate a predefined rule comprises the steps of: determining whether the first element is an interface and the second element is another

interface; and when it is determined that the first element is the interface and the second element is the other interface, identifying the link from the first element to the second element as an inheritance link.

Claim 92
The computer-readable medium of claim 91, wherein the method further comprises the step of identifying a link error when it is determined that the first element is the interface and the second element is not the other interface.

Claim 93
A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of elements, each element having corresponding code, the method comprising the steps of receiving a request to form a link; receiving an indication of a first of the plurality of elements; receiving an indication of a second of the plurality of elements; determining whether linking the first element to the second element would violate a predefined rule; and when it is determined that linking the first element to the second element would not violate a predefined rule, adding new code to the first element to reflect the link to the second element.

Claim 94
The computer-readable medium of claim 93, wherein the method further comprises the step of displaying a graphical representation of the code associated with the first element.

Claim 95
The computer-readable medium of claim 94, wherein the method further comprises the step of modifying the graphical representation of the code associated with the first element to reflect the link to the second element.

Claim 96
The computer-readable medium of claim 93, wherein the step of determining whether linking the first element to the second element would violate a predefined rule comprises the steps of: determining whether the first element is a class and whether the second element is another class; and when it is determined that the first element is the class and that the second element is the other class, identifying the link from the first element to the second element as an inheritance link.

Claim 97
The computer-readable medium of claim 93, wherein the step of determining whether linking the first element to the second element would violate a predefined rule comprises the steps of determining whether the first element is a class and whether the second element is an interface; and when it is determined that the first element is the class and that the second element is the interface, identifying the link from the first element to the second element as an implementation link.

Claim 98
The computer-readable medium of claim 93, wherein the step of determining whether linking the first element to second element would violate a predefined rule comprises the steps of determining whether the first element is an interface and the second element is another interface; and when it is determined that the first element is the interface and the second element is the other interface, identifying the link from the first element to the second element as an inheritance link.

Claim 99
A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of elements, each element having corresponding code, the method comprising the steps of: displaying a graphical representation of the code associated with a first of the plurality of elements and a graphical representation of the code associated with a second of the plurality of elements; receiving a request to form a link from the first element to the second element; determining whether the first element is a class and whether the second element is another class; and when it is determined that the first element is the class and that the second element is the other class, identifying the link from the first element to the second element as an inheritance link; adding new code to the first element to reflect the link to the second element; and modifying the graphical representation of the code associated with the first element to reflect the link to the second element.

Claim 100
The computer-readable medium of claim 99, wherein the method further comprises the steps of when it is determined that the first element is the class and that the second element is not the other class, determining whether the second element is an interface; and when it is determined that the second element is the interface, identifying the link from the first element to the second element as an implementation link.

Claim 101
The computer-readable medium of claim 99, wherein the method further comprises the steps of: when it is determined that the first element is not the class and that the second element is not the other class, determining whether the first and the second elements are interfaces; and when it is determined that the first and second elements are interfaces, identifying the link from the first element to the second element as an inheritance link.

Claim 102
A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of elements and having a link between two of the plurality of elements, wherein each element has corresponding code and the linked elements include a source and a destination, the method comprising the steps of: receiving a selection of one of the linked elements; receiving an identification of another of the plurality of elements that is different than the linked elements; determining whether the selected element is the destination; and when it is determined that the selected element is the destination, modifying

the corresponding code of the other element to reflect a new link between the other element and the destination element.

Claim 103
The computer-readable medium of claim 102, wherein the modifying step further includes the step of modifying the corresponding code of the source to reflect the removal of the link between the source and the destination.

Claim 104
The computer-readable medium of claim 103, wherein the method further comprises the step of displaying a graphical representation of the corresponding code of the source.

Claim 105
The computer-readable medium of claim 104, wherein the method further comprises the step of modifying the graphical representation of the corresponding code of the source to reflect the removal of the link between the source and the destination.

Claim 106
The computer-readable medium of claim 102, wherein the method further comprises the step of displaying a graphical representation of the corresponding code of the other element.

Claim 107
The computer-readable medium of claim 106, wherein the method further comprises the step of modifying the graphical representation of the corresponding code of the other element to reflect the new link between the other element and the destination
element.

Claim 108
The computer-readable medium of claim 102, wherein the modifying step includes the steps of determining whether linking the other element to the destination would violate a predefined rule; and when it is determined that linking the other element to the destination would not violate a predefined rule, modifying the corresponding code of the source to reflect the removal of the link between the source and the destination; and adding new code to the corresponding code of the other element to reflect the new link between the other element and the destination element.

Claim 109
The computer-readable medium of claim 108, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of: determining whether the other element is a class and whether the destination is another class; and when it is determined that the other element is the class and that the destination is the other class, identifying the new link between the other element and the destination as an inheritance link.

Claim 110

The computer-readable medium of claim 108, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of: determining whether the other element is a class and whether the destination is an interface; and when it is determined that the other element is the class and that the destination is the interface, identifying the new link between the other element and the destination as an implementation link.

Claim 111
The computer-readable medium of claim 110, wherein the method further comprises the step of identifying a link error when it is determined that the other element is the class and that the destination is not the interface.

Claim 112
The computer-readable medium of claim 108, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of determining whether the other element is an interface and the destination is another interface; and when it is determined that the other element is the interface and the destination is the other interface, identifying the new link between the other element and the destination as an inheritance link.

Claim 113
The computer-readable medium of claim 112, wherein the method further comprises the step of identifying a link error when it is determined that the other element is not the interface.

Claim 114
The computer-readable medium of claim 112, wherein the method further comprises the step of identifying a link error when it is determined that the destination is not the other interface.

Claim 115
The computer-readable medium of claim 102 wherein the method further comprises the step of: when it is determined that the selected element is the source, modifying the corresponding code of the source to reflect a new link between the source and the other element.

Claim 116
The computer-readable medium of claim 115, wherein the method further comprises the steps of: when it is determined that the selected element is the source, determining whether linking the source to the other element would violate a predefined rule; and when it is determined that linking the source to the other element would not violate a predefined rule, modifying the corresponding code of the source to reflect the removal of the link between the source and the destination; and adding new code to the corresponding code of the source to reflect the new link to the other element.

Claim 117

The computer-readable medium of claim 116, wherein the method further comprises the step of displaying a graphical representation of the code of the source to reflect the removal of the link and to reflect the new link to the other element when it is determined that the selected element is the source.

Claim 118
The computer-readable medium of claim 116, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of: determining whether the source is a class and whether the other element is another class; and when it is determined that the source is the class and that the other element is the other class, identifying the new link between the source and the other element as an inheritance link.

Claim 119
The computer-readable medium of claim 116, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of determining whether the source is a class and whether the other element is an interface; and when it is determined that the source is the class and that the other element is the interface, identifying the new link from the source to the other element as an implementation link.

Claim 120
The computer-readable medium of claim 119, wherein the method further comprises the step of identifying a link error when it is determined that the other element is not the interface.

Claim 121
The computer-readable medium of claim 116, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of determining whether the source is an interface and the other element is another interface; and when it is determined that the source is the interface and the other element is the other interface, identifying the new link between the source and the other element as an inheritance link.

Claim 122
The computer-readable medium of claim 121, wherein the method further comprises the step of identifying a link error when it is determined that the source is not the interface.

Claim 123
The computer-readable medium of claim 121, wherein the method further comprises the step of identifying a link error when it is determined that the other element is not the other interface.

Claim 124
A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of elements and having a link between two of the plurality of elements, wherein each element has corresponding code and the linked elements include a source and a destination, the method comprising the steps of: receiving

an identification of the link; receiving a selection of one of the linked elements; receiving an identification of another of the plurality of elements that is different than the linked elements; determining whether the selected element is the source; and when it is determined that the selected element is the source, modifying the corresponding code of the source to reflect a new link between the source and the other element.

Claim 125
The computer-readable medium of claim 124, wherein the method further comprises the step of displaying a graphical representation of the corresponding code of the source and a graphical representation of the corresponding code of the other element.

Claim 126
The computer-readable medium of claim 125, wherein the method further comprises the step of modifying the graphical representation of the corresponding code of the source to reflect the removal of the link when it is determined that the selected element is the source.

Claim 127
The computer-readable medium of claim 124, wherein the method further comprises the steps of: when it is determined that the selected element is the source, determining whether linking the source to the other element would violate a predefined rule; and when it is determined that linking the source to the other element would not violate a predefined rule, modifying the corresponding code of the source to reflect the removal of the link between the source and the destination; and adding new code to the corresponding code of the source to reflect the new link to the other element.

Claim 128
The computer-readable medium of claim 127, wherein the method further comprises the step of displaying a graphical representation of the code of the source to reflect the removal of the link and to reflect the new link to the other element when it is determined that the selected element is the source.

Claim 129
The computer-readable medium of claim 127, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of: determining whether the source is a class and whether the other element is another class; and when it is determined that the source is the class and that the other element is the other class, identifying the new link between the source and the other element as an inheritance link.

Claim 130
The computer-readable medium of claim 127, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of determining whether the source is a class and whether the other element is an interface; and when it is determined that the source is the class and that the other element is the interface, identifying the new link from the source to the other element as an implementation link.

Claim 131
The computer-readable medium of claim 130, wherein the method further comprises the step of identifying a link error when it is determined that the other element is not the interface.

Claim 132
The computer-readable medium of claim 127, wherein the step of determining whether linking the source to the other element would violate a predefined rule, comprises the steps of determining whether the source is an interface and the other element is another interface; and when it is determined that the source is the interface and the other element is the other interface, identifying the new link between the source and the other element as an inheritance link.

Claim 133
The computer-readable medium of claim 132, wherein the method further comprises the step of identifying a link error when it is determined that the source is not the interface.

Claim 134
The computer-readable medium of claim 132, wherein the method further comprises the step of identifying a link error when it is determined that the other element is not the other interface.

Claim 135
The computer-readable medium of claim 124, wherein the method further comprises the step of: when it is determined that the selected element is the destination, modifying the corresponding code of the other element to reflect a new link between the other element and the destination element.

Claim 136
The computer-readable medium of claim 135, wherein the modifying step further includes the steps of determining whether linking the other element to the destination would violate a predefined rule; and when it is determined that linking the other element to the destination would not violate a predefined rule, modifying the corresponding code of the source to reflect the removal of the link between the source and the destination; and adding new code to the corresponding code of the other element to reflect a new link between the other element and the destination element.

Claim 137
The computer-readable medium of claim 136, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of: determining whether the other element is a class and whether the destination is another class; and when it is determined that the other element is the class and that the destination is the other class, identifying the new link between the other element and the destination as an inheritance link.

Claim 138

The computer-readable medium of claim 136, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of determining whether the other element is a class and whether the destination is an interface; and when it is determined that the other element is the class and that the destination is the interface, identifying the new link between the other element and the destination as an implementation link.

Claim 139
The computer-readable medium of claim 138, wherein the method further comprises the step of identifying a link error when it is determined that the destination is not the interface.

Claim 140
The computer-readable medium of claim 136, wherein the step of determining whether linking the other element to the destination would violate a predefined rule, comprises the steps of: determining whether the other element is an interface and the destination is another interface; and when it is determined that the other element is the interface and the destination is the other interface, identifying the new link between the other element and the destination as an inheritance link.

Claim 141
The computer-readable medium of claim 140, wherein the method further comprises the step of identifying a link error when it is determined that the other element is not the interface.

Claim 142
The computer-readable medium of claim 140, wherein the method further comprises the step of identifying a link error when it is determined that the destination is not the other interface.

Claim 143
A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of elements and having a link between two of the plurality of elements, wherein each element has corresponding code and the linked elements include a source and a destination, the method comprising the steps of: displaying a graphical representation of the corresponding code of each of the plurality of elements; receiving a selection of one of the linked elements; receiving an identification of another of the plurality of elements that is different than the linked element; determining whether the selected element is the destination; and when it is determined that the selected element is the destination, determining whether the other element is a class and whether the destination is another class; and when it is determined that the other element is the class and that the destination is the other class, identifying a new link: from the other element to the destination as an inheritance link; removing a portion of the corresponding code of the source that reflects the link between the source and the destination; adding new code to corresponding code of the other element to reflect the new link between the other element and the destination; modifying the graphical representation of the corresponding code of the source to reflect the removal of the

link; and modifying the graphical representation of the corresponding code of the other element to reflect the new link.

Claim 144
The computer-readable medium of claim 143, wherein the method further comprises the steps of: when it is determined that the other element is the class and that the destination is not the other class, determining whether the destination is an interface; and when it is determined that the destination is the interface, identifying the new link between the other element and the destination as an implementation link.

Claim 145
The computer-readable medium of claim 143, wherein the method further comprises the steps of: when it is determined that the other element is not the class and that the destination is not the other class, determining whether the other element is an interface and whether the destination is another interface; and when it is determined that the other element is the interface and that the destination is the other interface, identifying the new link between the other element and the destination as an inheritance link

Claim 146
The computer-readable medium of claim 143, wherein the method further comprises the steps of when it is determined that the selected element is the source, determining whether the source is a class and whether the other element is another class; and when it is determined that the; source is the class and that the other element is the other class, identifying the new link between the source and the other element as an inheritance link; removing a portion of the corresponding code of the source that reflects the link between the source and the destination; and adding new code to corresponding code of the source to reflect the new link between the source and the other element.

Claim 147
The computer-readable medium of claim 146, wherein the method further comprises the step of modifying a graphical representation of the corresponding code of the source to reflect the removal of the link and to reflect the new link when it is determined that linking the source to the other element would not violate a predefined rule.

Claim 148
The computer-readable medium of claim 146, wherein the method further comprises the steps of: when it is determined that the source is the class and that the other element is not the other class, determining whether the other element is in an interface; and when it is determined that the other element is the interface, identifying the new link between the source and the other element as an implementation link.

Claim 149
The computer-readable medium of claim 146, wherein the method further comprises the steps of: when it is determined that the source is not the class and that the other element is not the other

class, determining whether the source is an interface and the other element is another interface; and when it is determined that the source is the interface and the other element is the other interface, identifying the new link between the source and the other element as an inheritance link.

Claim 150
A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of elements, the method comprising the steps of receiving an identification of a first of the plurality of elements; receiving an identification of a second of the plurality of elements; receiving an indication that the first element is to be included in the second element; determining whether the first element is a class and whether the second element is another class; and when it is determined that the first element is the class and that the second element is the other class, transferring code corresponding to the first element into the second element.

Claim 151
The computer-readable medium of claim 150, wherein the method further comprises the step of displaying a graphical representation of the code of the first element and a graphical representation of the code of the second element.

Claim 152
The computer-readable medium of claim 151, wherein the method further comprises the step of modifying the graphical representation of the code of the second element to reflect the transfer of the code corresponding to the first element into the second element.

Claim 153
The computer-readable medium of claim 150, wherein the step of transferring code comprises the steps of removing the code corresponding to the first element from a file; placing the code corresponding to the first element within the code corresponding to the second element; and deleting the file.

Claim 154
The computer-readable medium of claim 150, wherein the method further comprises the steps of when it is determined that the first element is the class and that the second element is not the other class, determining whether the second element is a package; and when it is determined that the second element is a package, moving a file that includes code corresponding to the first element to a directory associated with the second element.

Claim 155
A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of elements, each element having corresponding code, the method comprising the steps of: displaying a graphical representation of the code of a first of the plurality of elements and a graphical representation of the code of a second of the plurality of elements; receiving an indication that the first element is

to be included in the second element; determining whether the first element is a class and whether the second element is another class; and when it is determined that the first element is the class and that the second element is the other class, transferring code corresponding to the first element into the second element; and modifying a graphical representation of the code of the second element to reflect the transfer of the first element into the second element.

Claim 156
The computer-readable medium of claim 155, wherein the step of transferring code comprises the steps of: removing the code corresponding to the first element from a file; placing the code corresponding to the first element within code corresponding to the second element; and deleting the file.

Claim 157
A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of elements, each element having corresponding code, wherein code corresponding to a first of the plurality of elements is nested in the code corresponding to a second of the plurality of elements, the method comprising the steps of: receiving an indication that the first element is to be removed from the second element; determining whether the first element is a class and whether the second element is another class; and when it is determined that the first element is the class and that the second element is the other class, removing code corresponding to the first element from the second element.

Claim 158
The computer-readable medium of claim 157, wherein the method further comprises the step of placing the code corresponding to the first element into a file.

Claim 159
The computer-readable medium of claim 158, wherein the method further comprises the step of displaying a graphical representation of the code corresponding to the first element.

Claim 160
The computer-readable medium of claim 157, wherein the method further comprises the step of modifying a graphical representation of the code corresponding to the second element to reflect the removal of the first element from the second element.

Claim 161
The computer-readable medium of claim 157, wherein the method further comprises the steps of: when it is determined that the first element is the class and that the second element is not the other class, determining whether the second element is a package; and when it is determined that the second element is the package, removing a first file that includes code corresponding to the first element from a directory associated with the second element to another directory.

Claim 162

A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of elements, wherein a first of the plurality of elements is nested within a second of the plurality of elements, the method comprising the steps of: receiving an indication that the first element is to be removed from the second element; determining whether the first element is a class and whether the second element is another class; and when it is determined that the first element is the class and that the second element is the other class, removing code corresponding to the first element from the second element; placing the code corresponding to the first element into a file; and displaying a graphical representation of the code corresponding to the first element.

Claim 163
The computer-readable medium of claim 162, wherein the method further comprises the step of modifying a graphical representation of the code corresponding to the second element to reflect the removal of the first element from the second element.

Claim 164
The computer-readable medium of claim 162, wherein the method further comprises the steps of: when it is determined that the first element is the class and the second element is not the other class, determining whether the second element is a package; and when it is determined that the second element is the package, removing a first file that includes code corresponding to the first element from a directory associated with the second element.

Claim 165
A data processing system comprising: a secondary storage device further comprising a plurality of elements, each element having corresponding code; a memory device further comprising a program that receives a request to form a link, that receives an indication of a first of the plurality of elements, that receives an indication of a second of the plurality of elements, that determines whether linking the first element to the second element would violate a predefined rule, and when it is determined that linking the first element to the second element would not violate a predefined rule, the program adds new code to the first element to reflect the link to the second element; and a processor for running the program.

Claim 166
The data processing system of claim 165, wherein the program further displays a graphical representation of the code associated with the first element.

Claim 167
The data processing system of claim 166, wherein the program further modifies the graphical representation of the code associated with the first element to reflect the link to the second element.

Claim 168
The data processing system of claim 165, wherein when the program determines whether linking the first element to the second element would violate a predefined rule, the program determines

whether the first element is a class and whether the second element is another class, and when it: is determined that the first element is the class and that the second element is the other class, the program identifies the link from the first element to the second element as an inheritance link.

Claim 169
The data processing system of claim 165, wherein when the program determines whether linking the first element to the second element would violate a predefined rule, the program determines whether the first element is a class and whether the second element is an interface, and when it is determined that the first element is the class and that the second element is the interface, the program identifies the link from the first element to the second element as an implementation link.

Claim 170
The data processing system of claim 165, wherein when the program determines whether linking the first element to second element would violate a predefined rule, the program determines whether the first element is an interface and the second element is another interface, and when it is determined that the first element is the interface and the second element is the other interface, the program identifies the link from the first element to the second element as an inheritance link.

Claim 171
A data processing system comprising: a secondary storage device further comprising a plurality of elements and having a link between two of the plurality of elements, wherein each element has corresponding code and the linked elements include a source and a destination;
a memory device further comprising a program that displays a graphical representation of the corresponding code of each of the plurality of elements, that receives a selection of one of the linked elements, that receives an identification of another of the plurality of elements that is different than the linked element, that determines whether the selected element is the destination, and when it is determined that the selected element is the destination, the program determines whether the other element is a class and whether the destination is another class, and when it is determined that the other element is the class and that the destination is the other class, the program identifies a new link from the other element to the destination as an inheritance link, removes a portion of the corresponding code of the source that reflects the link between the source and the destination, adds new code to corresponding code of the other element to reflect the new link between the other element and the destination, modifies the graphical representation of the corresponding code of the source to reflect the removal of the link, and modifies the graphical representation of the corresponding code of the other element to reflect the new link; and a processor for running the program.

Claim 172
The data processing system of claim 171, wherein when it is determined that the other element is the class and that the destination is not the other class, the program further determines whether the destination is an interface, and when it is determined that the destination is the interface, the

program identifies the new link between the other element and the destination as an implementation link.

Claim 173
The data processing system of claim 171, wherein when it is determined that the other element is not the class and that the destination is not the other class, the program further determines whether the other element is an interface and whether the destination is another interface, and when it is determined that the other element is the interface and that the destination is the other interface, the program identifies the new link between the other element and the destination as an inheritance link.

Claim 174
The data processing system of claim 171, wherein when it is determined that the selected element is the source, the program further determines whether the source is a class and whether the other element is another class, and when it is determined that the source is the class and that the other element is the other class, the program identifies the new link between the source and the other element as an inheritance link, removes a portion of the corresponding code of the source that reflects the link between the source and the destination, and adds new code to corresponding code of the source to reflect the new link between the source and the other element.

Claim 175
The data processing system of claim 174, wherein the program further modifies a graphical representation of the corresponding code of the source to reflect the removal of the link and to reflect the new link when it is determined that linking the source to the other element would not violate a predefined rule.

Claim 176
The data processing system of claim 174, wherein when it is determined that the source is the class and that the other element is not the other class, the program further determines whether the other element is in an interface, and when it is determined that the other element is the interface, the program identifies the new link between the source and the other element as an implementation link.

Claim 177
The data processing system of claim 174, wherein when it is determined that the source is not the class and that the other element is not the other class, the program further determines whether the source is an interface and the other element is another interface, and when it is determined that the source is the interface and the other element is the other interface, the program identifies the new link between the source and the other element as an inheritance link.

Claim 178
A data processing system comprising: a secondary storage device further comprising a plurality of elements, each element having corresponding code; a memory device further comprising a

program that displays a graphical representation of the code of a first of the plurality of elements and a graphical representation of the code of a second of the plurality of elements, that receives an indication that the first element is to be included in the second element, that determines whether the first element is a class and whether the second element is another class, and when it is determined that the first element is the class and that the second element is the other class, the program transfers code corresponding to the first element into the second element, and modifies a graphical representation of the code of the second element to reflect the transfer of the first element into the second element; and a processor for running the program.

Claim 179
The data processing system of claim 178, wherein when the program transfers code, the program removes the code corresponding to the first element from a file, places the code corresponding to the first element within code corresponding to the second element, and deletes the file.

Claim 180
A data processing system comprising: a secondary storage device further comprising a plurality of elements, wherein a first of the plurality of elements is nested within a second of the plurality of elements; a memory device further comprising a program that receives an indication that the first element is to be removed from the second element, that determines whether the first element is a class and whether the second element is another class, and when it is determined that the first element is the class and that the second element is the other class, the program removes code corresponding to the first element from the second element, places the code corresponding to the first element into a file, and displays a graphical representation of the code corresponding to the first element; and a processor for running the program.

Claim 181
The data processing system of claim 180, wherein the program further modifies a graphical representation of the code corresponding to the second element to reflect the removal of the first element from the second element.

Claim 182
The data processing system of claim 180, wherein when it is determined that the first element is the class and the second element is not the other class, the program further determines whether the second element is a package, and when it is determined that the second element is the package, the program removes a first file that includes code corresponding to the first element from a directory associated with the second element.

Claim 183
A system having a plurality of elements, each element having corresponding code, the system comprising: means for receiving a request to form a link; means for receiving an indication of a first of the plurality of elements; means for receiving an indication of a second of the plurality of elements; and means for adding new code to the first element to reflect the link to the second element in response to receiving the request, the indication of the first element, and the indication of the second element.

7. In the broadest reasonable interpretation in view of the Specification the terms class/package and node are taught in the prior art. The Specification mentions UML but the claim language is silent.

Correspondence Information

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to **Todd Ingberg** whose telephone number is (703) 305-9775. The examiner can normally be reached during the following hours:

Monday	Tuesday	Wednesday	Thursday	Friday
6:15 – 1:30	6:15- 3:45	6:15 – 4:45	6:15-3:45	6:15-130

This schedule began December 1, 2003 and is subject to change.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, **Kakali Chaki** can be reached on (703) 305-9662. Please, note that as of August 4, 2003 the **FAX number** changed for the organization where this application or proceeding is assigned is **(703) 872-9306**.

Also, be advised the United States Patent Office **new address** is

Post Office Box 1450
Alexandria, Virginia 22313-1450

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-9700.

Todd Ingberg

Primary Examiner
Art Unit 2124
May 10, 2004